(~) tilde

encapsulation ⇒ إخفاء معلومات لا تهم المستخدم ووضعه في ملف خاص ويتم استخلاصه بـ header (.h)

implementation (تنفيذ)

في بداية البرامج — اسم الـ class

محفوظين
# ifndef _Date_h_ ⇒ if not define
# define _Date_h_
# include <iostream.h>
Class Date {
⋮

Constructor
→ default
→ overloading

};
# endif — في النهايات

دالة الـ main (في ملف ثاني 😊)

Source file
include <iostream.h>
include "Date.h" — Date مكتبة
int main()
{ Date d1;
⋮
}

محاضرة ٦:-

printf (format, variable)
↓ ↳ قائمة المتغيرات
توظيف المتغير

Q2 ⇒ Sheet # Struct

Functions ← ايجاد الزوايا :-
⇒ tan⁻¹ Θ
∴ 4 fns.

محاضرة ٦:- مهم

mult = multiplication (comp1, comp2); الزاوية
theta1 = atan (comp1.real/comp1.img) * (180/3.14); عشان
theta2 = atan (comp2.real / comp2.img) * (180/3.14); أجيب
theta3 = atan (add.real / add.img) * (180/3.14); السين
theta4 = atan (sub.real / sub.img) * (180/3.14);
theta5 = atan (mult.real / mult.img) * (180/3.14);

↗ img/real ← التصحيح

Printf (formate, variable)
ex. # لو متغير واحد
Printf ("Area = %d", area) ← decimal
%f Float
%c Character

لو أكثر من متغير التصحيح في حالات ⇐ %.3f ⇐ العشري في حالات
Printf ("%f length %f width %d Area", Length, width, Area);

ex.

```
class Square
{ private :
Attributes ← int side;
  public :
  Square() → Constructor with no parameters
Methods ← {side = 1; }                    نفس اسم ال class
};
```

```
class person          ② Destructors + Constructor ①
{ private:               مثل ال Methods
  int ID;                  في ال public
  String Name;    (و بنكتبهم في بداية ال method)   | person()  |
  int Age;                 (في ال UML)             | ~person() |
  public:           → هان منحطهم بس داخل
Person()  ←  void setID(int i)
{ID = 1; Name = "A"; Age = 18;}  { ID = i; }
destructor
~person()
{cout << "I'm destructor"; }
  int getID() → get ⇒ ارجاع قيمة
      {return ID; }                       move(?)
  void setName (String S)
      {Name = S; }
  String getName ()              inline function
      {return Name; }
  void SetAge (int a)
      { Age = a; }
  int getAge()
      {return Age; }
};
```

move (int x ...) move(?)
↓
increment ?

PAGE
DATE

```
class Square
{ private:
  int side;
  public:
  void setSide (int s);
  int getSide ();
};
void main() {
int Square :: getSide ()
{
  return side;
}
}
```

prototype

scope

```
ex. class A {
      int x →
      public:
      int get()
      {return x; }
    };
```

لو معملتش private هيتعامل معاها default value لأنها private

⇒ Constructors:- class لازم نفس اسم

default Constructors

لو معيش أرقام يعني = 1

- كل المتغيرات بتاعتي قيم
- كل المتغيرات ليها قيم
- قيمة ابتدائية واحدة

Invoking a Constructor

constructor استدعاء

Square square1;

Overloading

- قد يحتوي الـ class على أكثر من constructor

ex.

Square ();

Square (int); int x, int y

⇐ لازم parameters تكون مختلفة

class ⟵ مجموعة من ال objects تشترك في نفس الخصائص

◎ Attributes: int, String, ... etc

◎ Methods:

1. Constructor & destructor (أهم الذاكرة)
2. Modifier (Set fn.)
3. Access (Get fn.)
4. Other function (like display)

Unified Modelling Languages (UML)

Class Name
Attributes
Methods

يعادل ال Flowchart

ex. class ClassName (small letter: class) ~ Name (int a)

```
{
    declaration;
};
```

ex.

```
class Square
{ private:
    int side;
  public:
    void setSide (int s)
    { side = s; }              modifier
    int getSide ( )
    { return side; }           Access
};

void main ()
{ Square sq1, sq2;
  sq1.setSide(5);
  Cout << sq2.getSide ();
}
```

access specifiers ⟵ private / public

inline Function ⟵ الدوال داخل ال Class

Square
Side : int (اسم المتغير الأول ، النوع)
Set Side ( ) : void (اسم المتغير ، النوع)
get Side ( ) : int

كل ال set ما بعمله
ال get مرا بعمل

لو هعملهم prototype ⇐
وأكتبهم بره ال class
⇐ outline fn.

→ Enumerators عدّاد

الوظيفة : تجمع بعض المتغيرات التي يمكن حصرها

```
enum type {
    value 1,
    value 2;
} Object_name ;
```

ex.

```
enum Color {
  black, blue,
  green, red,
  purple, white
};
Color mycolor;
mycolor = red;   } → enum object
```

```
enum color {
  red = 3, green, blue,
  white = 8, black };
```


ex.

```
enum Months {
  January = 1, February, March, April, ...
} y2K;
```

خصصت أول متغير بقيمه = 1
هوتلقائياً هيكمل 2, 3, الخ لأنه عدّاد

بعض المصطلحات :-

→ Object Oriented (oop) object oriented programming

↳ وصف (تمثيل المعلومات) • متغيرات وان هناك متغيرات ال class

→ (Data encapsulation , information Hiding)

غطاء وإخفاء المعلومات لأنها تهم ال user

→ Data Abstraction

الوقت بين وصف المعلومات وتنفيذه

المبادئ الأساسية لل oop :-

\# Object # classes
\# encapsulation # Inheritance
\# poly

PAGE
DATE

```
cout << "sum = " << add.real << "+j" << add.img << endl;
cout << "sub = " << sub.real << "+j" << sub.img;
    return 0;
}
```

محاضرة 5:-

(Unions & Enumerators)

Union → داخل الـ Struct
تخزن قيمة واحدة فقط

```
Union WageInfo
{
  double hourlyRate;
};
```

→ Anonymous Union
Variable tag من غير Union

Structure with regular union:

```
Struct book {
Char title[50];
Char author[50];
Union {
  float dollars;
  int yen;
} price; ← variable
} book1;
```

Struct له (tag) هيأخد نفس اسم الـ Union

→ book1.price.dollars

Structure with anonymous Union:

```
Struct book {
Char title[50];
Char author[50];
Union {
  float dollars;
  int yen;
};
} book2;
```

→ book2.dollars

PAGE
DATE

Q2.

```
#include <iostream>
Using namespace std;
Struct Complex {
  float real, img;
};
Complex addition (Complex c1, Complex c2) definition
{
   Complex c;
   c.real = c1.real + c2.real;
   c.img = c1.img + c2.img;
    return c;
}
Complex subtraction (complex c1, complex c2)
 { Complex c;
    c.real = c1.real - c2.real;
    c.img = c1.img - c2.img;
     return c;
  }
  Void print (complex c)
    { cout << "(" << c.real << "," << c.img << ")"; }
  int main()
 { Complex c1,c2;
   Complex add, sub;
   cout << "Enter first complex No.";
   cin >> c1.real >> c1.img;
   cout << "Enter second complex No.";
     cin >> c2.real >> c2.img;
   add = addition (c1, c2);
   sub = subtraction (c1, c2);
   print (c1);
   print (c2);
```

calling ←

Sheet #3

Q1.

```
# include <iostream>
  Using namespace std;
  Struct Rectangle
  { float width, length;
   };
   float Area(float w, float L)        definition
  {                                    (;) عدم وجود ←
      return w*L;
   }
  float per (float w, float L)
  {
    return 2*(w+L);
  }
   Void print (float w, float L)
   {
   Cout<<"("<<w<<","<<L<<")"<<endl;  (5,6)
⇒   int main()                  (6,7)
   {
     Rectangle r;     الاسم ال tag بتاع ال struct بتعين
   Cout<<"Enter width and length of Rec :";
    cin>>r.width>>r.length;
     float area, perimeter;
    area= Area(r.width, r.length);  ← calling
     primeter =per(r.width, r.length);
    print (r.width, r.length);
     Cout<<"The area ="<<area;
     Cout<<"The perimeter="<<perimeter;
  }
```

نعرف ال struct خارج ال main

PAGE
DATE

⇒ Members of Nested Structures:-

```
Student S5;
S5.pData.name = "Ahmed";
S5.pData.city = "Tanta";
```

⇒ Passing members of struct to functions:-

```
ComputeGPA (S1.gpa);
```

⇒ Returning a structure from a function

(tag)

```
Student getStuData();   ⇒ prototype
S1();                   ⇒ calling
```

(variable)

~~int~~ x ( );

لو الأول والثالث {12,1,6};

```
Dimensions box1 = {12, 6};    Valid
Dimensions box2 = {12, , 3};  Invalid
```

كلهم واقفين عند الواحد default value

2 Constructor

1- يعامل معاملة ال member

2- inside the struct

```
Struct Dimensions
{ int length, width, height;
    Dimensions (int L=1, int W=1 , int H=1)
    { length = L; width = W; height = H; }
};
```

→ Struct of Constructor

```
① Dimensions box3 (12, 3, 5);
② Dimensions box4 (12, 6);
③ Dimensions box5;
```

لأنه default مش محتاج أكتبه وحده

→ Nested Structures

```
Struct PersonInfo
{ string name, address, city;
};
Struct Student
{ int StudentID;
    PersonInfo pData;    → هيخزن هنا ال name, address, city
    Short year;
    double gpa;
};
```

طرق الطباعة:-

Cout << s1; ✗ Invalid

→ Cout << s1.studentID;
Cout << s1.name;
Cout << s1.year;
Cout << s1.gpa;

→ أحدد المتغير التابع لـ s_1 (بحصري)

⇒ Comparing struct members:-

if (s1 >= s2) ✗ Invalid

→ if (s1.gpa >= s2.gpa)

⇒ Initializing a Structure — وضع القيم الابتدائية

int studentID = 1145; ✗ Invalid
String name = "Ali"; ✗ Invalid

مينفعش أدخل قيم ابتدائية على الـ Struct علطول

لـ بطريقتين:
1- initialization list
2- constructor

[1] Initialization list:

```
Struct Dimensions
{
   int length, width, height;
};
Dimensions box = {12, 6, 3};
```

(12 → length, 6 → width, 3 → height)

tag of the structure ← Dimensions

box ← اسم الـ var أي اسم شكل

محاضرة 4 :- م. تهاني

⇒ **Structure** :- is a C++ construct that allows multiple variables to be grouped together.

\# يسمح بتخزين أكثر من قيمة في الذاكرة

```
Struct  Structure_name
          ↳ tag: Capital يفضل يكون
  {   type1  field_1;
      type2  field_2;
       ...
      typen  field_n;
  };
```

Structure member ← (type1 field_1; … typen field_n;)

```
Student  S1;        ⇒ main جوة ال
```

S1 → أي متغير عادي

or

```
Struct Student
 {
   int StudentID;
   String name;
   Short year;
   double gpa;
 } S1;
```

S1 → variable

⇒ **Accessing Structure members:**

\# طرق مختلفة لإدخال القيم :-

```
   Cout << "Enter the Student name :";
①  getline ( Cin , S1. name);
   Cout << "Enter the Student ID :";
②  cin >> S1. StudentID;
③  S1.gpa = 3.75;
```

(S1 → Struct variable ← [illegible])

```
// function definition
void dbl_By_Address (int * a_ptr)
{ *a_ptr = 2 * (*a_ptr);
}
```

⇒ output

i before passing its pointer to the function = 1

i after execution of the function = 8

* Swapping variable values :-

تبديل قيم المتغيرات

3 برامج

```
# include <iostream.h>
  void swapInt (int *, int *);
  int main()
{ int i=35, j=50;
  Cout << "In Before swapping : i= " << i << "j=" << j;
   SwapInt (&i, &j);
  Cout << "In After swapping : i=" << i << "j=" << j;
   Cout << endl;
   return 0;
}
   Void swapInt (int * a_ptr, int * b_ptr)
   { int temp;
     temp = *a_ptr;
     *a_ptr = *b_ptr;
     *b_ptr = temp;
   }
```

→ Output :-

Before swapping : i = 35, j = 50

After swapping : i = 50, j = 35

PAGE
DATE

```
cout<< endl;
 return 0;
```

⇒ Output :-

Enter the name of the salesperson : Ahmed Ibrahim
Enter the sales for Monday : 650.6
" " " Tuesday : 438.9
" " " Wednesday : 321.5
" " " Thursday : 750.8
" " " Friday : 862.2
" " " Saturday : 651.0
" " " Sunday = 411.8
The total sales for Ahmed Ibrahim are 4086.80.
The highest sales were 862.20.
The highest sales occured on Friday.

Pointers & Arrays & functions

استدعاء دالة لمضاعفة قيمتها عن طريق الـ address

برنامج 2

```
#include <iostream.h>
 void db1_By_Address (int *);
 int main()
{ int i=4;
 cout << "\n i before passing its pointer to the function:" << i;
   db1_By_Address(&i);
 cout << "\n i after the execution of the function =" << i;
   cout << endl;
   return 0;
```

Marketings

#١٤ اكتب برنامج لحساب مبيعات شخص وتحديد أقصى المبيعات واليوم الذي حدث له ذلك

برنامج 1

```
# include <iostream.h>
  int main()
 {
   Char  dayNames [7][10] = { "Monday", "Tuesday",
             "Wednesday", "Thursday", "Friday", "Saturday", "Sunday" };
   double  sales [7] ; →            لو Char هيكون 8 لوجود "\0"
   Char  salesperson [41]; [40 +1]                      أو String
   int day, maxDay ;
   Cout << "\n Enter the name of the salesperson :";
   Cin.getline ( salesperson, 41 ) ;
   for ( day = 0 ; day <7 ; ++day )
   { Cout << "\n\n Enter the sales for" << dayNames [day] << ":";
      Cin>> sales >> sales [day] ;
   }
      total sales = 0 ; maxDay = 0 ;  max Sales = Sales [0] ;
   for ( day = 0 ; day <7 ; ++day )        (<=) لأنها ++day
    { if ( sales [day] > maxsales )           وليس day++
        {    max sales = Sales [day] ;
             maxDay = day ;
        }
         total Sales += sales [day] ;
    }
      Cout << "\n\n The total sales for" << Sales person <<
           "are" << total Sales << "." ;
      Cout << "\n\n The highest sales were" << max sales <<
     Cout << "\n\n The highest Sales occurred on " <<
         day Names [maxday] << "." ;
```

PAGE
DATE

محاضرة 3 :-

المصفوفة المكونة من سلاسل حرفية .

Array Consists of strings of Characters :-

Char cityNames [6][9] = { "Cairo", "Amman", "New York", 2D
"London", "Beirut", "Khartoum" };

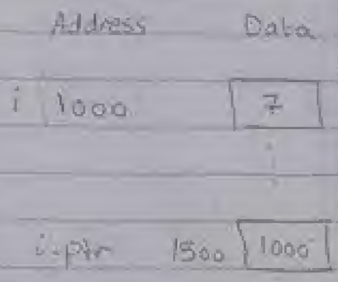
C	a	i	r	o	\0			
A	m	m	a	n	\0			
N	e	w		Y	o	r	K	\0
L	o	n	d	o	n	\0		
B	e	i	r	u	t	\0		
K	h	a	r	t	o	o	m	\0

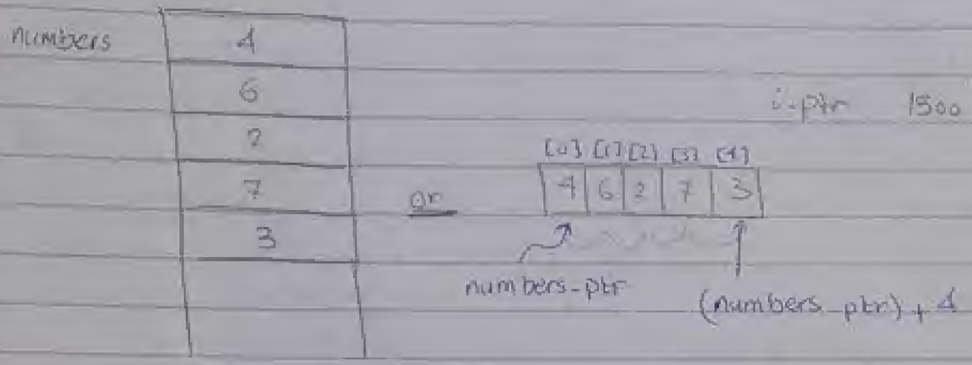
1D أو

	C
	a
	i
	r
	o
	\0
1	
2	
3	
	A
	m
	m

⇒ Pointer

```
int numbers[5] = {4, 6, 2, 7, 3};
int *numbers_ptr;
```





بوينتر ← متغير يشير إلى عنوان متغير آخر من نفس النوع

⇒ Pointer and strings:-

لو حرف ← 'T';   5 حروف و ال 6 ← \0 ⟷ أسيبها فاضية أحسن

```
Char mycity[6] = "Tanta";
```

mycity | T | a | n | t | a | \0 |

⇒ String input & output

```
Cin.getline (name, size);
```

size +1 (null عشان ال \0)

```
getline (cin, mystring);
```
?!

PAGE
DATE

```
cin >> A[i][j];
}
Cout << "\n Output matrix array";
for(i=0; i<3; i++)        للصفوف
{ Cout << "\n";
   for(j=0; j<4; j++)     للأعمدة
   Cout << "\t" << A[i][j];
}
return 0;
}
```

⇒ Output

Enter element of array

A[0][0] = 3
A[0][1] = 4
A[0][2] = 5
A[0][3] = 6
A[1][0] = 0
A[1][1] = 2
A[1][2] = 4
A[1][3] = 5
A[2][0] = 3
A[2][1] = 2
A[2][2] = 8
A[2][3] = 0

Output matrix aray

3 4 5 6
0 2 4 5
3 2 8 0

← ممكن نجمع أو نضرب كل عنصر في 2 مثلاً

↓

مصفوفة جديدة ⇒ Mat[i][j] = 2* A[i][j];

يتم تعريفها في بداية البرنامج .

PAGE
DATE

محاضرة 2 :-

Write a program to enter an array with (2D) Using keyboard and print it as follows :-

	0	1	2	3
0	3	4	5	6
1	0	2	4	5
2	3	2	8	0

⇒ Algorithm

* Start
* loop 1 i i=0 to n step 1
  ~~input A[i][j]~~ ~~End loop 1~~ → loop j j=0 to m step 1
  input A[i][j]
  end loop j
* loop 2 i=0 to n step 1
  loop 3 j=0 to m step 1
  Output A[i][j]
  End loop 3.
  End loop 2
* End

⇒ The program

```
/* ---------
   ---------*/
#include <iostream>
int main()
{ int A[3][4];
  int i,j;
  Cout << "Enter elements of array \n";
  for(i=0; i<3; i++)  للصفوف
{ for (j=0; j<4; j++)  للأعمدة
  Cout << "A[" <<i<<"][" <<j<< "]=";
```

... برمجة الحاسب ...

محاضرة 1 :-

macro ⇒ هي دالة من سطر واحد

1 Byte = 8 Bits

تحديد قيم الخانات (من غير Keyboard)

```
for( ... )
cin >> x
```
(crossed out: X)

Arrays :-

int ID [5];

2	-3	5	-15	99

ID [0] = 2;

⋮

ID [4] = 99;

or

int ID [5] = { 2, -3, 5, -15, 99 };

	0	1	2	3
0	2	-3	5	-15
1	1	2	4	-10

int temp [2][4] = { { 2, -3, 5, -15 }, { 1, 2, 4, -10 } };

(عدد الصفوف ↓ , عدد الاعمدة →)

or

temp [0][0] = 2;

⋮

temp [1][3] = -10;